Characterizing License Practices in Maven Central and Their Relationship with CVEs: A CS846 Course Report

4/3/25

Haonan Zhang, Christina Li, Paul Wooseok Lee

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

Motivation & Background

- Open-source software is central to modern development.
- Maven Central hosts hundreds of thousands of artifacts.
- Licenses determine usage, reuse, and redistribution.
- Vulnerability (CVE) reporting is essential for security risk management.

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

What is the Relationship between Licenses and CVE Patterns?

Hypothesis

- Certain license types may influence collaboration and patching processes, potentially correlating with higher or lower vulnerability rates.

Objective

- Systematically analyze license data and CVEs in Maven Central to uncover patterns that can guide more informed artifact and license adoption choices.

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

Current Studies Primarily Focus on Either License or CVEs

Studies on license:


A Large-Scale Empirical Study of Open Source License Usage: Practices and Challenges

Jiaqi Wu, The State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, Zhejiang, China, jiaqiwu@zju.edu.cn
Lingfeng Bao*, The State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, Zhejiang, China, lingfengbao@zju.edu.cn
Xiaohu Yang, The State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, Zhejiang, China, yangxh@zju.edu.cn
Xin Xia, Huawei, China, xin.xia@acm.org
Xing Hu, The State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, Zhejiang, China, xinghu@zju.edu.cn

Investigating whether and how software developers understand open source software licensing

Daniel A. Almeida[1] · Gail C. Murphy[1] · Greg Wilson[2] · Michael Hoye[3]

On the Adoption of Open Source Software Licensing - A Pattern Collection

Petr Pícha, Department of Computer Science and Engineering, University of West Bohemia, Czechia, ppicha@kiv.zcu.cz
Souhaila Serbout, Software Institute (USI), Switzerland, souhaila.serbout@usi.ch

Studies on CVEs:


Understanding the Threats of Upstream Vulnerabilities to Downstream Projects in the Maven Ecosystem

Yulun Wu[1†], Zeliang Yu[1†], Ming Wen[1*], Qiang Li[2], Deqing Zou[2], Hai Jin[2]
[1]School of Cyber Science and Engineering, Huazhong University of Science and Technology, China
[2]School of Computer Science and Technology, Huazhong University of Science and Technology, China
Email: {alanwu, zeliangyu, mwenas, lujiang, deqingzou, hjin}@hust.edu.cn


Mitigating Persistence of Open-Source Vulnerabilities in Maven Ecosystem

Lyuye Zhang*, Chengwei Liu[†], Sen Chen[‡], Zhengzi Xu[†], Lingling Fan[§], Lida Zhao[¶], Yiran Zhang[†], Yang Liu[†]
[*][†]Continental-NTU Corporate Lab, Nanyang Technological University, Singapore
[‡]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[§]College of Intelligence and Computing, Tianjin University, China
[¶]College of Cyber Science, Nankai University, China


Analyzing the Direct and Transitive Impact of Vulnerabilities onto Different Artifact Repositories

JOHANNES DÜSING and BEN HERMANN, Technical University Dortmund, Germany

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

Research Questions

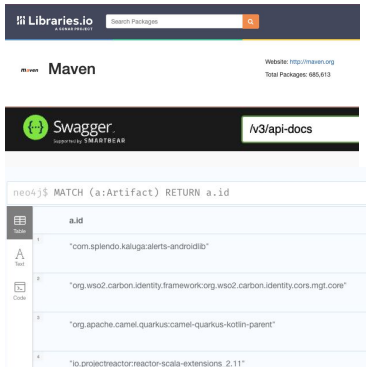
RQ1:

- What are the characteristics and trends of license adoption and CVE incidence across Maven Central artifacts?

RQ2:

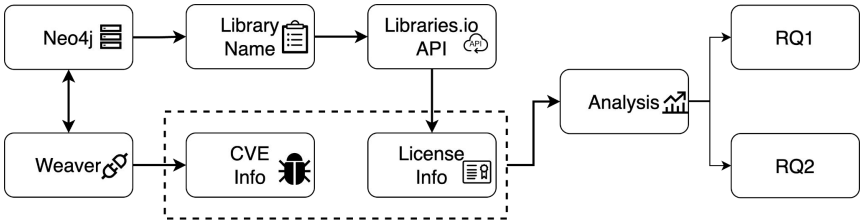
- Do specific license types correlate with higher or lower vulnerability incidence in Maven Central artifacts?

Dataset and Data Extraction



- **Data sourced from Maven Central, Libraries.io, Neo4j, and Weaver API.**
- **Approximately 278,984 records extracted from a subset (over 3 days) out of 658K records.**
- **Data split into batches and processed via Python scripts using JSON formats.**

Tools and Methodology



License Extraction and Ranking

```
from collections import Counter

# Count occurrences of each normalized license
license_counter = Counter(rec["norm_license"] for rec in records)
# Get top 10 by occurrence
top_licenses = license_counter.most_common(10)
# Exclude "other" (case-insensitive) and extract only the license names
top_license_names = [lic for lic, count in top_licenses if lic.lower() != "other"]
print("Top normalized licenses (excluding 'other'):")
for lic in top_license_names:
    print(f"{lic}: {license_counter[lic]}")
```

```
Top normalized licenses (excluding 'other'):
apache-2.0: 169439
mit: 46853
gpl-3.0: 7392
epl-2.0: 5220
lgpl-3.0: 4975
bsd-3-clause: 4545
lgpl-2.1+: 3089
epl-1.0: 3198
mulanpsl-2.0: 3086
```

```
def deduplicate_and_select_top(records, top_k=100):
    """
    Given a list of records, deduplicates them by the library's 'name' field.
    For each library name, only the record with the highest dependents_count is kept.
    Returns a list of records sorted by dependents_count (descending), limited to top_k.
    """
    dedup = {}
    for rec in records:
        lib_name = rec["name"]
        # If already seen, keep the one with higher dependents_count.
        if lib_name in dedup:
            if rec["dependents_count"] > dedup[lib_name]["dependents_count"]:
                dedup[lib_name] = rec
        else:
            dedup[lib_name] = rec
    dedup_list = list(dedup.values())
    dedup_list.sort(key=lambda x: x["dependents_count"], reverse=True)
    return dedup_list[:top_k]

final_records = []
seen_libs = set()  # Global set to track library names already added.

for lic in top_license_names:
    if lic in groups:
        top_records = deduplicate_and_select_top(groups[lic], top_k=100)
        for rec in top_records:
            # Only add if the library name hasn't been added before.
            if rec["name"] not in seen_libs:
                final_records.append({
                    "normalized_license": rec["norm_license"],
                    "name": rec["name"],
                    "dependents_count": rec["dependents_count"]
                })
                seen_libs.add(rec["name"])

print(f"Total final records (across top licenses, unique libraries): {len(final_records)}")
```

Integrating CVE Data

```python
import json
import requests

def get_cve_info(library_name, start_ts, end_ts, host="http://localhost:8080"):
    """
    Constructs and sends a query to the Weaver API to get CVE info for the given library.
    The query looks for nodes where n.id starts with library_name and with timestamp in the specified range.
    Returns the JSON response from the Weaver API.
    """
    query_str = (
        f"MATCH (n) WHERE n.id STARTS WITH '{library_name}' "
        f"AND n.timestamp >= {start_ts} AND n.timestamp <= {end_ts} RETURN n"
    )
    payload = {
        "query": query_str,
        "addedValues": ["CVE"]
    }
    try:
        response = requests.post(f"{host}/cypher", json=payload)
        if response.status_code == 200:
            return response.json()
        else:
            print(f"Error for library {library_name}: {response.status_code} {response.text}")
            return None
    except Exception as e:
        print(f"Exception for library {library_name}: {e}")
        return None

def main():
    # Define timestamp range (milliseconds)
    start_ts = 1584489600000  # e.g., March 18, 2020 00:00:00 UTC
    end_ts = 1735689600000    # e.g., January 1, 2025 00:00:00 UTC
```

- CVE information retrieved via the Weaver API with cypher queries.
- Queries parameterized by library names allow integration of vulnerability data with library metadata.
- Data processed and aggregated using Python (json, collections.Counter).

Answering RQ1:

RQ1:

- What are the characteristics and trends of license adoption and CVE incidence across Maven Central artifacts?

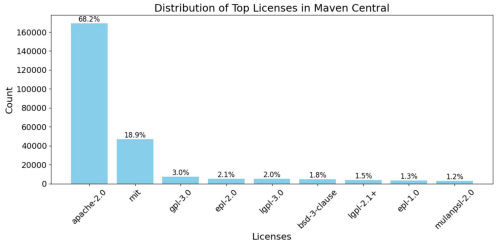
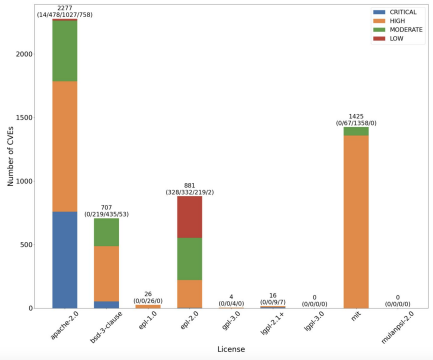


Table 1. Number of releases of 100 top libraries per license

License Name	Total Releases
apache-2.0	30,822
bsd-3-clause	4,276
epl-1.0	8,743
epl-2.0	6,143
gpl-3.0	6,893
lgpl-2.1+	5,405
lgpl-3.0	3,764
mit	15,011
mulanpsl-2.0	4,050

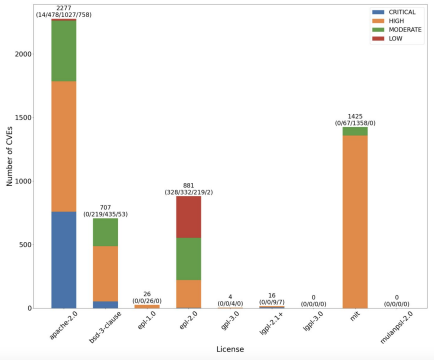
Answering RQ1 Cont'd



Finding 1:

- Artifacts under permissive licenses like the Apache-2.0 exhibit a **notably high** number of CVEs.
- In contrast, licenses with stronger copyleft provisions, which have lower adoption rates, report **fewer** vulnerabilities.

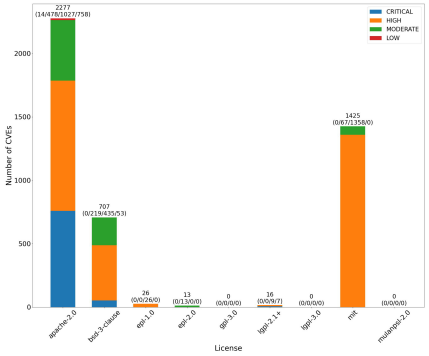
Answering RQ1 Cont'd



Finding 2:

- MulanPSL-2.0 has **zero** number of CVEs as a permissive license.
- However, EPL-2.0, has **high** number of CVEs as a restrictive license.

Answering RQ1 Cont'd



Finding 3:

- After removing libraries with both permissive and restrictive licenses for all licenses, EPL-2.0 went from **881 CVEs to 13 CVEs**.

Answering RQ1 Cont'd

License Name	Total Releases
apache-2.0	30,822
bsd-3-clause	4,276
epl-1.0	8,743
epl-2.0	6,143
gpl-3.0	6,893
lgpl-2.1+	5,405
lgpl-3.0	3,764
mit	15,011
mulanpsl-2.0	4,050

License Name	Release Number
apache-2.0	30,822
bsd-3-clause	4,276
epl-1.0	8,628
epl-2.0	1,546
gpl-3.0	5,981
lgpl-2.1+	5,311
lgpl-3.0	3,198
mit	15,011
mulanpsl-2.0	4,050

Finding 3:

- After removing libraries with both permissive and restrictive licenses for all licenses, EPL-2.0 went from **881 CVEs to 13 CVEs**.

Answering RQ2:

RQ2: Do specific license types **correlate with higher or lower** vulnerability incidence in Maven Central artifacts?

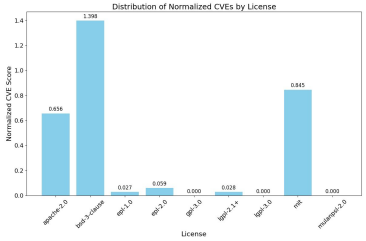


Table 2. Updated number of releases after removing libraries with both permissive and restrictive licenses

License Name	Release Number
apache-2.0	30,822
bsd-3-clause	4,276
epl-1.0	8,628
epl-2.0	1,546
gpl-3.0	5,981
lgpl-2.1+	5,311
lgpl-3.0	3,198
mit	15,011
mulanpsl-2.0	4,050

Table 3. CVSS Severity Rating

Metric Value	Severity Level
0-4	Low
4-7	Moderate
7-9	High
9-10	Critical

Answering RQ2 Cont'd: Statistical test

Mann-Whitney U test:

- We choose the Mann-Whitney U test because it does not enforce any assumptions about the distribution of analyzed data and applies to a small number of data points.
- **p = 0.036**

Finding 4:

- The permissive licenses exhibit higher CVE incidence, and the Mann-Whitney U test further confirms that there is a statistically significant difference between permissive and restrictive licenses. Consequently, **our data provides evidence of a correlation between license type and vulnerability incidence.**

Threats to Validity

- Limited extraction (278K records out of 658K) might introduce selection bias.
- Analysis focused only on top 10 licenses and top 100 libraries per license, which may overlook less popular libraries.
- Temporal restrictions on CVE data (2020–2025) may not fully capture historical vulnerability trends.
- Weaver API's unidirectional query (from library to CVE) limits comprehensive vulnerability capture.

Future Work

1. Use full dataset to conduct analysis, whether through improved data extracting efficiency or devote more time into the process.
2. Expand analysis to include more license types and alternative ranking metrics.
3. Extend temporal range to capture longer CVE history.
4. Enhance API capabilities to support reverse querying (from CVE to libraries) for deeper insights.
5. Future studies could delve deeper into the implications of libraries that adopt multiple licenses, particularly those mixing permissive and restrictive types.

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

UNIVERSITY OF WATERLOO

FACULTY OF MATHEMATICS